Brown clusters, linguistic context, and spectral algorithms

Daniel Hsu

Columbia University

Joint work with

Mike Collins (Columbia/Google)  Do-kyum Kim (Google)  Karl Stratos (Columbia)
1. Introduction
Learning from unlabeled data

Many applications of machine learning

- Lots of high-dimensional data.
- Mostly unlabeled—i.e., not annotated with prediction target.
Learning from unlabeled data

Many applications of machine learning
▶ Lots of high-dimensional data.
▶ Mostly unlabeled—i.e., not annotated with prediction target.

What kinds of structure can we learn from unlabeled data?
Examples from natural language processing

- **Example 1:** Language models

\[
\frac{P(\text{colorless green ideas sleep furiously})}{P(\text{furiously sleep ideas green colorless})} \gg 1
\]
Examples from natural language processing

▶ **Example 1**: Language models

\[
\frac{P(\text{colorless green ideas sleep furiously})}{P(\text{furiously sleep ideas green colorless})} \gg 1
\]

▶ **Example 2**: Word sense disambiguation

\begin{align*}
(\text{“bank”}, \{\text{“stocks”, “bonds”, …} \}) \\
\text{vs.} \quad (\text{“bank”}, \{\text{“river”, “freshwater”, …} \})
\end{align*}
Examples from natural language processing

▶ **Example 1**: Language models

\[
P(\text{colorless green ideas sleep furiously}) \\
\frac{P(\text{furiously sleep ideas green colorless})}{P(\text{furiously sleep ideas green colorless})} \gg 1
\]

▶ **Example 2**: Word sense disambiguation

\[
(\text{“bank”, \{“stocks”, “bonds”, …\}}) \\
\text{vs. } (\text{“bank”, \{“river”, “freshwater”, …\}})
\]

Doesn’t require any direct supervision to learn!
Examples from natural language processing

► Example 1: Language models

\[
P(\text{colorless green ideas sleep furiously}) \quad \gg \quad P(\text{furiously sleep ideas green colorless})
\]

► Example 2: Word sense disambiguation

\[
(\text{“bank”, \{“stocks”, “bonds”, . . . \}}) \\
\text{vs. } (\text{“bank”, \{“river”, “freshwater”, . . . \}})
\]

► Example 3: “Word classes”

e.g., \{“apple”, “pear”, . . . \}, \{“Apple”, “IBM”, . . . \},
\{“bought”, “run”, . . . \}, \{“of”, “in”, . . . \}, . . .

Doesn’t require any direct supervision to learn!

- **Brown clustering**: clustering a vocabulary into **word classes** using the Brown clustering algorithm

<table>
<thead>
<tr>
<th>class 1</th>
<th>class 2</th>
<th>class 3</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>feet</td>
<td>people</td>
<td>water</td>
<td></td>
</tr>
<tr>
<td>miles</td>
<td>guys</td>
<td>gas</td>
<td></td>
</tr>
<tr>
<td>pounds</td>
<td>folks</td>
<td>coal</td>
<td></td>
</tr>
<tr>
<td>degrees</td>
<td>fellows</td>
<td>liquid</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Q: What do these word classes capture?

Not entirely clear, but...

- **Brown clustering**: clustering a vocabulary into word classes using the Brown clustering algorithm

<table>
<thead>
<tr>
<th>class 1</th>
<th>class 2</th>
<th>class 3</th>
<th>⋮</th>
</tr>
</thead>
<tbody>
<tr>
<td>feet</td>
<td>people</td>
<td>water</td>
<td></td>
</tr>
<tr>
<td>miles</td>
<td>guys</td>
<td>gas</td>
<td></td>
</tr>
<tr>
<td>pounds</td>
<td>folks</td>
<td>coal</td>
<td></td>
</tr>
<tr>
<td>degrees</td>
<td>fellows</td>
<td>liquid</td>
<td></td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
</tbody>
</table>

Q: What do these word classes capture?
Word class models


- **Brown clustering**: clustering a vocabulary into word classes using the Brown clustering algorithm

<table>
<thead>
<tr>
<th>class 1</th>
<th>class 2</th>
<th>class 3</th>
<th>⋮</th>
</tr>
</thead>
<tbody>
<tr>
<td>feet</td>
<td>people</td>
<td>water</td>
<td></td>
</tr>
<tr>
<td>miles</td>
<td>guys</td>
<td>gas</td>
<td></td>
</tr>
<tr>
<td>pounds</td>
<td>folks</td>
<td>coal</td>
<td></td>
</tr>
<tr>
<td>degrees</td>
<td>fellows</td>
<td>liquid</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td></td>
</tr>
</tbody>
</table>

Q: What do these word classes capture? Not entirely clear, but …
Semi-supervised Natural Language Processing:

1. Apply Brown clustering to large corpus of unlabeled text to derive "lexical representations" (a.k.a. word representations).
2. Augment existing NLP methods with lexical representations.
3. Win!

- Named-entity recognition (Miller et al., 2004; Turian et al., 2010)
- Dependency parsing (Koo et al., 2008)
- Language modeling (Kneser and Ney, 1993; Gao et al., 2001)

Our goal: Understand & build on the success of Brown clustering
Semi-supervised Natural Language Processing:

1. Apply Brown clustering to large corpus of unlabeled text to derive “lexical representations” (a.k.a. word representations).
Semi-supervised Natural Language Processing:

1. Apply Brown clustering to large corpus of unlabeled text to derive “lexical representations” (a.k.a. word representations).

2. Augment existing NLP methods with lexical representations.
Structure from a language model

Semi-supervised Natural Language Processing:

1. Apply Brown clustering to large corpus of unlabeled text to derive “lexical representations” (a.k.a. word representations).
2. Augment existing NLP methods with lexical representations.
3. Win!
   - Named-entity recognition (Miller et al, 2004; Turian et al, 2010)
   - Dependency parsing (Koo et al, 2008)
   - Language modeling* (Kneser and Ney, 1993; Gao et al, 2001)
   - ...
Structure from a language model

Semi-supervised Natural Language Processing:

1. Apply Brown clustering to large corpus of unlabeled text to derive “lexical representations” (a.k.a. word representations).
2. Augment existing NLP methods with lexical representations.
3. Win!
   - Named-entity recognition (Miller et al., 2004; Turian et al., 2010)
   - Dependency parsing (Koo et al., 2008)
   - Language modeling* (Kneser and Ney, 1993; Gao et al., 2001)
   - ... 

Our goal: Understand & build on the success of Brown clustering
Our contributions

Motivating observation: Learning Brown word classes only requires correlations between words & simple linguistic context.
Our contributions

Motivating observation: Learning Brown word classes only requires correlations between words & simple linguistic context.

What we do:

1. Propose a spectral algorithm for learning word classes in the setting of Brown et al [Stratos, Kim, Collins, & H, UAI 2014]
   - Algorithmically simple, amenable to theoretical analysis
   - Empirically faster than Brown clustering algorithm

   - Theoretically understood in Brown et al setting
   - Improves lexical representations for low-level NLP tasks

3. Assess ability of Brown word class model to capture real linguistic structure—real test of unsupervised learning [Stratos, Collins, & H, TACL 2016]
Our contributions

**Motivating observation:** Learning Brown word classes only requires **correlations between words & simple linguistic context.**

**What we do:**

1. Propose a spectral algorithm for learning word classes in the setting of Brown *et al* [Stratos, Kim, Collins, & H, UAI 2014]
   - Algorithmically simple, amenable to theoretical analysis
   - Empirically faster than Brown clustering algorithm

   - Theoretically understood in Brown *et al* setting
   - Improves lexical representations for low-level NLP tasks
Our contributions

Motivating observation: Learning Brown word classes only requires correlations between words & simple linguistic context.

What we do:

1. Propose a spectral algorithm for learning word classes in the setting of Brown et al [Stratos, Kim, Collins, & H, UAI 2014]
   - Algorithmically simple, amenable to theoretical analysis
   - Empirically faster than Brown clustering algorithm

   - Theoretically understood in Brown et al setting
   - Improves lexical representations for low-level NLP tasks

3. Assess ability of Brown word class model to capture real linguistic structure—real test of unsupervised learning. [Stratos, Collins, & H, TACL 2016]
Talk outline

1. Spectral algorithm for learning word classes in the setting of Brown et al
   [Stratos, Kim, Collins, & H, UAI 2014]

2. Improved estimation using variance stabilization
   [Stratos, Collins, & H, ACL 2015]

3. Using Brown word class model for unsupervised POS tagging
   [Stratos, Collins, & H, TACL 2016]
2. Examining the Brown word class model
The Brown et al word class model (parameters)

HMM with hidden state seq. \( (H_t) \) and observation seq. \( (X_t) \).
The Brown et al word class model (parameters)

HMM with hidden state seq. \((H_t)\) and observation seq. \((X_t)\).

Hidden state space = word classes \(C := \{1, 2, \ldots, |C|\}\).
Observation space = vocabulary \(V := \{1, 2, \ldots, |V|\}\).
Column-stochastic parameters \(\theta := (\pi, T, O)\)

\[
\begin{align*}
\pi_h &= P_\theta[H_1 = h], & h \in C, \\
T_{g,h} &= P_\theta[H_{t+1} = g \mid H_t = h], & (g, h) \in C \times C, \\
O_{x,h} &= P_\theta[X_t = x \mid H_t = h], & (x, h) \in V \times C.
\end{align*}
\]
The Brown et al word class model (structural restriction)

Brown et al word class model places structural restriction on $O$:

There is a hard clustering of vocabulary $V$ into $|C|$ groups

$\{V_h : h \in C\}$ (the word classes) such that

$$x \in V_h \implies P_\theta[X_t = x \mid H_t = g] = 0 \text{ for all } g \neq h.$$ 

Each word can be generated by the hidden state corresponding to its word class.

![Sparsity pattern of emission probability matrix $O$](image)

(after permuting rows)
Log-likelihood in the word class model

Max-likelihood parameters that respect clustering $C$ is (up to const.)
empirical mutual information bet. word classes of adjacent words

\[
\sum_t \sum_{g,h} \hat{\Pr}[C(X_t) = g, C(X_{t+1}) = h] \ln \frac{\hat{\Pr}[C(X_t) = g, C(X_{t+1}) = h]}{\hat{\Pr}[C(X_t) = g] \hat{\Pr}[C(X_{t+1}) = h]}.
\]

Under the Brown word class model:
max log-likelihood $\Leftrightarrow$ max $\hat{\text{MIs}}$ between classes of adjacent words
Log-likelihood in the word class model

Max-likelihood parameters that respect clustering $C$ is (up to consts.) empirical mutual information bet. word classes of adjacent words

$$
\sum_t \sum_{g,h} \hat{Pr}[C(X_t) = g, C(X_{t+1}) = h] \ln \frac{\hat{Pr}[C(X_t) = g, C(X_{t+1}) = h]}{\hat{Pr}[C(X_t) = g] \hat{Pr}[C(X_{t+1}) = h]}
$$

Under the Brown word class model:
max log-likelihood $\Leftrightarrow$ max $\hat{M}_1$s between classes of adjacent words

Not clear how to efficiently maximize w.r.t. clustering $C$. 
Log-likelihood in the word class model

Max-likelihood parameters that respect clustering \( C \) is (up to consts.)
empirical mutual information bet. word classes of adjacent words

\[
\sum_t \sum_{g,h} \hat{\Pr}[C(X_t) = g, C(X_{t+1}) = h] \ln \frac{\hat{\Pr}[C(X_t) = g, C(X_{t+1}) = h]}{\hat{\Pr}[C(X_t) = g] \hat{\Pr}[C(X_{t+1}) = h]}
\]

Under the Brown word class model:
max log-likelihood \( \Leftrightarrow \) max \( \hat{\text{MIs}} \) between classes of adjacent words

Not clear how to efficiently maximize w.r.t. clustering \( C \).

**Brown clustering algorithm** (Brown et al, 1992):

- Start with each word in its own class.
- Repeat: merge class pair that decreases \( \hat{\text{MIs}} \) the least.

Output: a *hierarchy* of word classes.
Output of Brown clustering algorithm

```
Output of Brown clustering algorithm

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

apple  pear  Apple  IBM  bought  run  of  in

Use in NLP: augmenting text data with lexical representations increases ability for (supervised) ML methods to learn other linguistic structure.
Output of Brown clustering algorithm

Get **lexical representations** from a pruning of the hierarchy:

<table>
<thead>
<tr>
<th>word</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple</td>
<td>00</td>
</tr>
<tr>
<td>pear</td>
<td>00</td>
</tr>
<tr>
<td>Apple</td>
<td>01</td>
</tr>
<tr>
<td>IBM</td>
<td>01</td>
</tr>
<tr>
<td>bought</td>
<td>10</td>
</tr>
<tr>
<td>run</td>
<td>10</td>
</tr>
<tr>
<td>of</td>
<td>11</td>
</tr>
<tr>
<td>in</td>
<td>11</td>
</tr>
</tbody>
</table>

Use in NLP: augmenting text data with lexical representations increases ability for (supervised) ML methods to learn other linguistic structure.
Get **lexical representations** from a pruning of the hierarchy:

- apple → 00
- pear → 00
- Apple → 01
- IBM → 01
- bought → 10
- run → 10
- of → 11
- it → 11

**Use in NLP:** augmenting text data with lexical representations increases ability for (supervised) ML methods to learn other linguistic structure.
Our aim: extract word classes directly from observable quantities.

Theorem (Stratos, Kim, Collins, and H., 2014)

Define matrix $B \in \mathbb{R}^{V \times V}$, $B_{x, y} := n^{-1} \sum_{t=1}^{n} P(\theta | X_t = x, X_{t+1} = y)$. If data follow a Brown model distribution, then left singular vectors of $B$ reveal the word classes (after row normalization). $B$ can be estimated directly from raw collection of sentences.
Word classes from observable quantities

**Our aim:** extract word classes directly from observable quantities.

**Theorem (Stratos, Kim, Collins, and H, 2014)**

*Define matrix* $\mathbf{B} \in \mathbb{R}^{V \times V}$

$$B_{x,y} := \sum_{t=1}^{n-1} P_\theta(X_t = x, X_{t+1} = y).$$

If data follow a Brown model distribution, then left singular vectors of $\mathbf{B}$ “reveal” the word classes (after row normalization).

$U = \begin{array}{c}
\text{purple} \\
\text{green} \\
\text{orange} \\
\end{array}$

$Q$ = (rotated rows)

$Q = \begin{array}{c}
\text{purple} \\
\text{green} \\
\text{orange} \\
\end{array}$

(rotated rows)
Word classes from observable quantities

Our aim: extract word classes directly from observable quantities.

Theorem (Stratos, Kim, Collins, and H, 2014)

Define matrix $B \in \mathbb{R}^{V \times V}$

$$B_{x,y} := \sum_{t=1}^{n-1} P_\theta(X_t = x, X_{t+1} = y).$$

If data follow a Brown model distribution, then left singular vectors of $B$ “reveal” the word classes (after row normalization).
Word classes from observable quantities

**Our aim:** extract word classes directly from observable quantities.

**Theorem (Stratos, Kim, Collins, and H, 2014)**

*Define matrix* \( B \in \mathbb{R}^{V \times V} \)

\[
B_{x,y} := \sum_{t=1}^{n-1} P_{\theta}(X_t = x, X_{t+1} = y).
\]

*If data follow a Brown model distribution, then left singular vectors of* \( B \) *“reveal” the word classes (after row normalization).*

*\( B \) can be estimated directly from raw collection of sentences.*
1. Form estimate $\hat{B}$ of $B$ matrix, e.g.,

$$\hat{B}_{x,y} := \sum_{t=1}^{n-1} \hat{Pr}(X_t = x, X_{t+1} = y),$$

and compute its rank-$|C|$ thin SVD $\hat{U}\hat{S}\hat{V}^\top$.

2. For each $x \in V$, let $q_x$ be the corresponding row in $\hat{U}$, normalized to have unit length.

3. Apply agglomerative clustering (e.g., average-linkage) to vectors $\{q_x : x \in V\}$.

Output: a hierarchy of word classes.

**Bonus:** Main computational bottleneck (SVD) is a well-studied numerical linear algebra problem with highly-optimized solutions.
Spectral algorithm for Brown clustering

1. Form estimate $\hat{B}$ of $B$ matrix, e.g.,

$$\hat{B}_{x,y} := \sum_{t=1}^{n-1} \hat{\Pr}(X_t = x, X_{t+1} = y),$$

and compute its rank-$|C|$ thin SVD $\hat{U}\hat{S}\hat{V}^\top$.

2. For each $x \in V$, let $q_x$ be the corresponding row in $\hat{U}$, normalized to have unit length.

3. Apply agglomerative clustering (e.g., average-linkage) to vectors $\{q_x : x \in V\}$.

Output: a hierarchy of word classes.

**Bonus**: Main computational bottleneck (SVD) is a well-studied numerical linear algebra problem with highly-optimized solutions.
Improvements

Context  \( X_{t+1} \) is “linguistic context” for \( X_t \).
Can also use richer context
e.g., \((X_{t-2}, X_{t-1}, X_{t+1}, X_{t+2})\)
(two words before, two words after).
**Improvements**

**Context**  \( X_{t+1} \) is “linguistic context” for \( X_t \).

Can also use richer context
e.g., \((X_{t-2}, X_{t-1}, X_{t+1}, X_{t+2})\)
(two words before, two words after).

**Transforms** Main Theorem holds even if we apply certain linear transformations to \( B \).

Does not change core structural properties,
but may improve conditioning.
Empirical study

Both Brown clustering and spectral algorithm provide (hierarchy of) word classes.

Questions:

1. How does spectral algorithm compare to Brown clustering on Brown clustering objective ($\hat{\text{MI}}$ between adjacent classes)?

2. How does spectral algorithm compare to Brown clustering in utility of lexical representations?
Question 1: Brown clustering objective

**Data:** RCV1 news articles (205M tokens).

**Method:** Compare Brown clustering with Spectral algorithm, both with $|C| = 1000$ classes.

| Algorithm | $|V|$ | $\hat{MI}$ | Time |
|-----------|------|-----------|------|
| Spectral  | 50K  | 1.48      | 0.37h|
|           | 300K | 1.54      | 2.07h|
| Brown     | 50K  | 1.52      | 3.62h|
|           | 300K | 1.60      | 22.33h|
Question 2: Utility of lexical representations

**Data:** News articles for CoNLL 2003 Named Entity Recognition shared task.

**Method:** Using $|C| = 1000$ lexical representations from RCV1, with Perceptron + greedy decoding (Ratinov and Roth, 2009). (Standard semi-supervised approach to this NLP problem.)

| Algorithm | $|V|$ | dev F1 | test F1 |
|-----------|------|--------|---------|
| Baseline  |      | 90.03  | 84.39   |
| Spectral  | 50K  | 92.00  | 86.72   |
|           | 300K | 92.31  | 87.76   |
| Brown     | 50K  | 92.00  | 88.56   |
|           | 300K | 92.68  | 88.76   |

(There is known discrepancy between dev & test sets here.)
Observations

- Spectral algorithm much faster than Brown clustering in terms of wall-clock time (up to $10 \times$ speed-up).
Observations

- Spectral algorithm much faster than Brown clustering in terms of wall-clock time (up to 10× speed-up).

- Spectral algorithm lags Brown clustering in terms of Brown clustering objective (MI).

- Major limitation: Hard-clustering forces a class to capture all senses of any member word.

- Possible fix: Skip the clustering step! Directly use representation given by left singular vectors of $\hat{B}$. 
Observations

- Spectral algorithm much faster than Brown clustering in terms of wall-clock time (up to $10 \times$ speed-up).

- Spectral algorithm lags Brown clustering in terms of Brown clustering objective (MI).

- Both algorithms provide lexical representations that deliver comparable improvements over baseline.
Observations

- Spectral algorithm much faster than Brown clustering in terms of wall-clock time (up to 10× speed-up).
- Spectral algorithm lags Brown clustering in terms of Brown clustering objective (MI).
- Both algorithms provide lexical representations that deliver comparable improvements over baseline.

**Major limitation:** Hard-clustering forces a class to capture all senses of any member word.
Observations

- Spectral algorithm much faster than Brown clustering in terms of wall-clock time (up to $10 \times$ speed-up).

- Spectral algorithm lags Brown clustering in terms of Brown clustering objective ($\hat{\text{MI}}$).

- Both algorithms provide lexical representations that deliver comparable improvements over baseline.

**Major limitation**: Hard-clustering forces a class to capture all senses of any member word.

**Possible fix**: Skip the clustering step! Directly use representation given by left singular vectors of $\hat{B}$. 
3. Dealing with noise heteroskedasticity
Motivation

- Main estimation task in spectral algorithm is estimating word/context pairs frequencies $B$
  (more specifically, the left singular vectors of $B$).

- How can we do better on this estimation task?

- **Challenge**: many word/context pairs have very different frequencies, and hence very different “estimation noise variance”.
Basic spectral algorithm

**Simplified setting**: word is $X$, context is $Y$. 
Basic spectral algorithm

**Simplified setting:** word is $X$, context is $Y$.

**Basic spectral algorithm:**

- Use raw co-occurrence counts from $N$ sentences

$$\hat{B}_{x,y} := \#(X = x, Y = y)$$

(ignoring normalization).

- Decompose into low-rank factors using SVD, i.e., minimize

$$\min_{L \in \mathbb{R}^{V \times C}, R \in \mathbb{R}^{V \times C}} \| LR^\top - \hat{B} \|_F^2.$$
Possible improvement

Since “noise” \( \hat{B} - B \) is highly heteroskedastic, could be better to minimize variance-normalized squared error

\[
\min_{L \in \mathbb{R}^{V \times C}, \quad R \in \mathbb{R}^{V \times C}} \sum_{x,y} \frac{1}{\text{var}(\hat{B}_{x,y})} \left((LR^T)_{x,y} - \hat{B}_{x,y}\right)^2.
\]

(C.f. weighted least squares.)
Possible improvement

Since “noise” $\hat{B} - B$ is highly heteroskedastic, could be better to minimize variance-normalized squared error

$$\min_{L \in \mathbb{R}^{V \times C}, \quad R \in \mathbb{R}^{V \times C}} \sum_{x,y} \frac{1}{\text{var}(\hat{B}_{x,y})} \left( (LR^T)_{x,y} - \hat{B}_{x,y} \right)^2.$$  

(C.f. weighted least squares.)

However, weighted objective is hard to minimize (Srebro et al, 2003).
A statistical trick

**Square-root trick:** Instead of using $\hat{B}$, use $\sqrt{\hat{B}}$ (element-wise square-root of $\hat{B}$).
A statistical trick

Square-root trick: Instead of using $\hat{B}$, use $\sqrt{\hat{B}}$ (element-wise square-root of $\hat{B}$).

Asymptotic justification:

- Poisson approximation: when $p_{x,y} := \Pr(X = x, Y = y)$ is small compared to $1/N$, approximately have

$$\hat{B}_{x,y} \sim \text{Poi}(N \cdot p_{x,y}).$$
A statistical trick

Square-root trick: Instead of using $\hat{B}$, use $\sqrt{\hat{B}}$ (element-wise square-root of $\hat{B}$).

Asymptotic justification:

- **Poisson approximation:** when $p_{x,y} := \Pr(X = x, Y = y)$ is small compared to $1/N$, approximately have

  $$\hat{B}_{x,y} \sim \text{Poi}(N \cdot p_{x,y}).$$

- **Variance stabilization:** As $N \to \infty$,

  $$\text{var}(\sqrt{\hat{B}_{x,y}}) \to 1/4$$

  (Bartlett, 1936; Anscombe, 1948).
Variance stabilization

A heuristic derivation via delta method:
For $g(x) := \sqrt{x}$ and $X \sim \text{Poi}(\lambda)$,

$$g(X) \approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X))$$
Variance stabilization

A heuristic derivation via delta method:
For $g(x) := \sqrt{x}$ and $X \sim \text{Poi}(\lambda)$,

$$g(X) \approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X))$$

$$= \sqrt{\lambda} + \frac{1}{2\sqrt{\lambda}} \cdot (X - \lambda).$$
Variance stabilization

A heuristic derivation via delta method:
For \( g(x) := \sqrt{x} \) and \( X \sim \text{Poi}(\lambda) \),

\[
g(X) \approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X)) \\
= \sqrt{\lambda} + \frac{1}{2 \sqrt{\lambda}} \cdot (X - \lambda).
\]

Therefore

\[
\text{var}(g(X)) \approx \left( \frac{1}{2 \sqrt{\lambda}} \right)^2 \cdot \text{var}(X)
\]
Variance stabilization

A heuristic derivation via delta method:
For \( g(x) := \sqrt{x} \) and \( X \sim \text{Poi}(\lambda) \),

\[
g(X) \approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X)) \\
= \sqrt{\lambda} + \frac{1}{2\sqrt{\lambda}} \cdot (X - \lambda).
\]

Therefore

\[
\text{var}(g(X)) \approx \left( \frac{1}{2\sqrt{\lambda}} \right)^2 \cdot \text{var}(X) \\
= \frac{1}{4\lambda} \cdot \lambda
\]
Variance stabilization

A heuristic derivation via delta method:
For \( g(x) := \sqrt{x} \) and \( X \sim \text{Poi}(\lambda) \),

\[
g(X) \approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X))
\]

\[
= \sqrt{\lambda} + \frac{1}{2\sqrt{\lambda}} \cdot (X - \lambda).
\]

Therefore

\[
\text{var}(g(X)) \approx \left( \frac{1}{2\sqrt{\lambda}} \right)^2 \cdot \text{var}(X)
\]

\[
= \frac{1}{4\lambda} \cdot \lambda = \frac{1}{4}.
\]
Variance stabilization

A heuristic derivation via delta method:
For $g(x) := \sqrt{x}$ and $X \sim \text{Poi}(\lambda)$,

$$g(X) \approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X))$$

$$= \sqrt{\lambda} + \frac{1}{2\sqrt{\lambda}} \cdot (X - \lambda).$$

Therefore

$$\text{var}(g(X)) \approx \left(\frac{1}{2\sqrt{\lambda}}\right)^2 \cdot \text{var}(X)$$

$$= \frac{1}{4\lambda} \cdot \lambda = \frac{1}{4}.$$

So asymptotically, don’t need variance normalization.
Variance stabilization

A heuristic derivation via delta method:
For \( g(x) := \sqrt{x} \) and \( X \sim \text{Poi}(\lambda) \),

\[
g(X) \approx g(\mathbb{E}(X)) + g'(\mathbb{E}(X)) \cdot (X - \mathbb{E}(X))
\]

\[
= \sqrt{\lambda} + \frac{1}{2\sqrt{\lambda}} \cdot (X - \lambda).
\]

Therefore

\[
\text{var}(g(X)) \approx \left( \frac{1}{2\sqrt{\lambda}} \right)^2 \cdot \text{var}(X)
\]

\[
= \frac{1}{4\lambda} \cdot \lambda = \frac{1}{4}.
\]

So asymptotically, don’t need variance normalization.

Moreover, using \( \sqrt{\widehat{B}} \) make senses in the Brown model:
Left singular vectors of \( \sqrt{B} \) also reveal word classes, just like \( B \)’s.
Empirical study

**Question**: Does the Brown word class model capture the same intrinsic qualities as other popular lexical representations?
Empirical study

**Question**: Does the Brown word class model capture the same intrinsic qualities as other popular lexical representations?

▶ **Synonyms**: How well do cosine similarities between lexical representations reflect human judgements?

```
argmax \ x \in \ V \langle q_x, q_{Australia} \rangle - \langle q_x, q_{Canberra} \rangle + \langle q_x, q_{London} \rangle.
```
Empirical study

**Question:** Does the Brown word class model capture the same intrinsic qualities as other popular lexical representations?

▶ **Synonyms:** How well do cosine similarities between lexical representations reflect human judgements?

▶ **Analogies:** How well do lexical representations provide answer to analogy problems like

\[
\text{Canberra is to Australia, as London is to ______}
\]

based on cosine similarities:

\[
\arg \max_{x \in V} \langle q_x, q_{\text{Australia}} \rangle - \langle q_x, q_{\text{Canberra}} \rangle + \langle q_x, q_{\text{London}} \rangle.
\]
Empirical study

**Question**: Does the Brown word class model capture the same intrinsic qualities as other popular lexical representations?

- **Synonyms**: How well do cosine similarities between lexical representations reflect human judgements?

- **Analogies**: How well do lexical representations provide answer to analogy problems like

  \[
  \text{Canberra is to Australia, as London is to ______}
  \]

  based on cosine similarities:

  \[
  \arg \max_{x \in V} \langle q_x, q_{Australia} \rangle - \langle q_x, q_{Canberra} \rangle + \langle q_x, q_{London} \rangle.
  \]

  Are these measures predictive of utility in extrinsic tasks?
Data sources

- **Training data**: English Wikipedia, 1.4B tokens.
Data sources

- **Training data**: English Wikipedia, 1.4B tokens.

- **Similarity tasks**: Agirre et al.’s “WordSim353”, Bruni et al.’s “MEN Test Collection”, and Stanford Rare Word Similarity Dataset: (5.4K word pairs + human assessments)
  
  Measure Pearson correlation with human assessments.
Data sources

- **Training data**: English Wikipedia, 1.4B tokens.

- **Similarity tasks**: Agirre et al.’s “WordSim353”, Bruni et al.’s “MEN Test Collection”, and Stanford Rare Word Similarity Dataset: (5.4K word pairs + human assessments)
  
  Measure Pearson correlation with human assessments.

- **Analogy tasks**: Microsoft (Mikolov-Yih-Zweig) dataset of “syntactic” analogies: (8000 questions)
  
  Google (Mikolov et al) dataset of “syntactic” and “semantic” analogies: 19544 questions

  Measure word prediction accuracy.
Results

Other methods (with same context $X_{t-2}, X_{t-1}, X_{t+1}, X_{t+2}$ as Spectral):

- Continous bag-of-words (Mikolov et al, 2013) in Word2Vec
- Skip-gram (Mikolov et al, 2013) in Word2Vec
- PPMI (Levy and Goldberg, 2014)
- Glove (Pennington, Socher, Manning, 2014)

<table>
<thead>
<tr>
<th>Method</th>
<th>dimension = 500</th>
<th></th>
<th></th>
<th>dimension = 1000</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SIM  (corr)</td>
<td>MSFT (acc%)</td>
<td>GOOG (acc%)</td>
<td>SIM  (corr)</td>
<td>MSFT (acc%)</td>
<td>GOOG (acc%)</td>
</tr>
<tr>
<td>Spectral</td>
<td>0.572</td>
<td>39.68</td>
<td>57.64</td>
<td>0.650</td>
<td>66.08</td>
<td>76.38</td>
</tr>
<tr>
<td>Spectral$\sqrt{+}$</td>
<td>0.655</td>
<td>68.38</td>
<td>74.17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBOW</td>
<td>0.597</td>
<td>75.79</td>
<td>73.60</td>
<td>0.509</td>
<td>70.97</td>
<td>60.12</td>
</tr>
<tr>
<td>SKIP</td>
<td>0.642</td>
<td>81.08</td>
<td>78.73</td>
<td>0.641</td>
<td>79.98</td>
<td>83.35</td>
</tr>
<tr>
<td>PPMI</td>
<td>0.628</td>
<td>43.81</td>
<td>58.38</td>
<td>0.637</td>
<td>48.99</td>
<td>63.82</td>
</tr>
<tr>
<td>Glove</td>
<td>0.576</td>
<td>68.30</td>
<td>78.08</td>
<td>0.586</td>
<td>67.40</td>
<td>78.73</td>
</tr>
</tbody>
</table>
Utility in extrinsic tasks

Directly use vectors $q_x$ as features in structured prediction for Named Entity Recognition (again, CoNLL 2003 shared task).

<table>
<thead>
<tr>
<th>Method</th>
<th>30 dimensions</th>
<th>50 dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dev F1</td>
<td>test F1</td>
</tr>
<tr>
<td>Baseline</td>
<td>90.03</td>
<td>84.39</td>
</tr>
<tr>
<td>Brown</td>
<td>92.49</td>
<td>88.75</td>
</tr>
<tr>
<td><strong>Spectral</strong> $+\sqrt{-}$</td>
<td><strong>92.88</strong></td>
<td><strong>89.28</strong></td>
</tr>
<tr>
<td>CBOW</td>
<td>92.44</td>
<td>88.34</td>
</tr>
<tr>
<td>SKIP</td>
<td>92.63</td>
<td>88.78</td>
</tr>
<tr>
<td>PPMI</td>
<td>92.25</td>
<td>89.27</td>
</tr>
<tr>
<td>Glove</td>
<td>91.49</td>
<td>87.16</td>
</tr>
</tbody>
</table>

(There is known discrepancy between dev & test sets here.)
Utility in extrinsic tasks

Directly use vectors $q_x$ as features in structured prediction for Named Entity Recognition (again, CoNLL 2003 shared task).

<table>
<thead>
<tr>
<th>Method</th>
<th>30 dimensions</th>
<th>50 dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dev F1</td>
<td>test F1</td>
</tr>
<tr>
<td>Baseline</td>
<td>90.03</td>
<td>84.39</td>
</tr>
<tr>
<td>Brown</td>
<td>92.49</td>
<td>88.75</td>
</tr>
<tr>
<td><strong>Spectral</strong></td>
<td>92.88</td>
<td><strong>89.28</strong></td>
</tr>
<tr>
<td>CBOW</td>
<td>92.44</td>
<td>88.34</td>
</tr>
<tr>
<td>SKIP</td>
<td>92.63</td>
<td>88.78</td>
</tr>
<tr>
<td>PPMI</td>
<td>92.25</td>
<td>89.27</td>
</tr>
<tr>
<td>Glove</td>
<td>91.49</td>
<td>87.16</td>
</tr>
</tbody>
</table>

(There is known discrepancy between dev & test sets here.)

All improve over baseline; **Spectral** is computationally cheapest.
Observations

- Spectral performs well on similarity tasks, less competitive on analogy tasks.
- Poor analogy performance doesn’t seem to hurt much for extrinsic NER task.
Observations

- Spectral performs well on similarity tasks, less competitive on analogy tasks.
- Poor analogy performance doesn’t seem to hurt much for extrinsic NER task.

**Question**: Which extrinsic tasks are analogy-adept lexical representations especially good for?
4. Unsupervised learning
What linguistic structure is captured by HMM?
Capturing linguistic structure without supervision

What linguistic structure is captured by HMM?

- **Hypothesis**: Parts-of-Speech (e.g., noun, verb, adj)

Evidence: running (unsupervised) EM, initialized at HMM learned with supervision, only makes things worse.

Upshot: Do not use likelihood to test the hypothesis.
Capturing linguistic structure without supervision

What linguistic structure is captured by HMM?

- **Hypothesis:** Parts-of-Speech (e.g., noun, verb, adj)

- **Test:** Do word classes correspond to parts-of-speech?
  Learn word class model, then measure “many-to-one accuracy”, using *true* labels of words (e.g., from a dictionary).
What linguistic structure is captured by HMM?

- **Hypothesis**: Parts-of-Speech (e.g., noun, verb, adj)

- **Test**: Do word classes correspond to parts-of-speech?
  Learn word class model, then measure “many-to-one accuracy”, using *true* labels of words (e.g., from a dictionary).

- **Folklore**: maximum likelihood HMM is unlikely to yield states that correspond to parts-of-speech.
What linguistic structure is captured by HMM?

- **Hypothesis**: Parts-of-Speech (e.g., noun, verb, adj)
- **Test**: Do word classes correspond to parts-of-speech?
  Learn word class model, then measure “many-to-one accuracy”, using *true* labels of words (e.g., from a dictionary).
- **Folklore**: maximum likelihood HMM is unlikely to yield states that correspond to parts-of-speech.

  *Evidence*: running (unsupervised) EM, initialized at HMM learned with supervision, only makes things worse.
What linguistic structure is captured by HMM?

- **Hypothesis**: Parts-of-Speech (e.g., noun, verb, adj)
- **Test**: Do word classes correspond to parts-of-speech?

  Learn word class model, then measure “many-to-one accuracy”, using *true* labels of words (e.g., from a dictionary).

- **Folklore**: maximum likelihood HMM is unlikely to yield states that correspond to parts-of-speech.

  *Evidence*: running (unsupervised) EM, initialized at HMM learned with supervision, only makes things worse.

- **Upshot**: Do not use likelihood to test the hypothesis.
Instead of likelihood, exploit linguistic context.

- Find HMM consistent with linguistic context (e.g., surrounding words) and features (e.g., spelling features).
Instead of likelihood, exploit linguistic context.

- Find HMM consistent with linguistic context (e.g., surrounding words) and features (e.g., spelling features).

**Our approach:**

- Use spectral algorithm to derive lexical representation vectors.
- Apply farthest-first traversal to these vectors to pick “anchors”.
- Use Bayes’ rule + convex optimization to estimate HMM parameters (previously proposed by Arora-Ge-Moitra, 2012).
Some results

Data from universal treebank, 12 POS tag types

<table>
<thead>
<tr>
<th>Method</th>
<th>de</th>
<th>en</th>
<th>es</th>
<th>fr</th>
<th>id</th>
<th>it</th>
<th>ja</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-M</td>
<td>45.5</td>
<td>59.8</td>
<td>60.6</td>
<td>60.1</td>
<td>49.6</td>
<td>51.5</td>
<td>59.5</td>
</tr>
<tr>
<td>Brown</td>
<td>60.0</td>
<td>62.9</td>
<td>67.4</td>
<td>66.4</td>
<td>59.3</td>
<td>66.1</td>
<td>60.3</td>
</tr>
<tr>
<td>Spectral⁺√</td>
<td>61.1</td>
<td>66.1</td>
<td>69.0</td>
<td>68.2</td>
<td>63.7</td>
<td>60.4</td>
<td>65.3</td>
</tr>
<tr>
<td>Spectral⁺√+f</td>
<td>63.4</td>
<td>71.4</td>
<td>74.3</td>
<td>71.9</td>
<td>67.3</td>
<td>60.2</td>
<td>69.4</td>
</tr>
<tr>
<td>Log-linear (Berg-Kirkpatrick et al, 2010)</td>
<td>67.5</td>
<td>62.4</td>
<td>67.1</td>
<td>62.1</td>
<td>61.3</td>
<td>52.9</td>
<td>78.2</td>
</tr>
</tbody>
</table>

- Spectral⁺√ = just use prev/next words context.
- Spectral⁺√+f = also uses spelling features.
- Log-linear (Berg-Kirkpatrick et al, 2010): not a HMM
Final remarks

- Yet more confirmation that linguistic context is very powerful:
Final remarks

- Yet more confirmation that linguistic context is very powerful:
  “We” already knew the information was there; just need algorithmic/statistical techniques to fully exploit it.
Final remarks

- Yet more confirmation that linguistic context is very powerful: “We” already knew the information was there; just need algorithmic/statistical techniques to fully exploit it.

- Brown et al word class model is surprisingly simple — an obviously “wrong” model, but captures a lot of useful structure.
Final remarks

- Yet more confirmation that linguistic context is very powerful: “We” already knew the information was there; just need algorithmic/statistical techniques to fully exploit it.
- Brown et al word class model is surprisingly simple — an obviously “wrong” model, but captures a lot of useful structure.
- Unclear what is the “right” intrinsic evaluation of lexical representations.
Final remarks

- Yet more confirmation that linguistic context is very powerful: “We” already knew the information was there; just need algorithmic/statistical techniques to fully exploit it.

- Brown et al word class model is surprisingly simple — an obviously “wrong” model, but captures a lot of useful structure.

- Unclear what is the “right” intrinsic evaluation of lexical representations.

Thank you!
Anchor word assumption (Arora, Ge, Moitra, 2012) is strictly weaker than assumption in Brown word class model.

- For each hidden state $h \in C$, there is an “anchor” word $x \in V$ satisfying
  $$O_{x,g} = 0 \text{ for all } g \neq h.$$
Connection to anchor word assumption

**Anchor word assumption** (Arora, Ge, Moitra, 2012) is strictly weaker than assumption in Brown word class model.

- For each hidden state $h \in C$, there is an “anchor” word $x \in V$ satisfying

  $$O_{x,g} = 0 \text{ for all } g \neq h.$$

- For comparison, Brown model word class assumption requires every word to be an “anchor”.
Connection to anchor word assumption

Anchor word assumption (Arora, Ge, Moitra, 2012) is strictly weaker than assumption in Brown word class model.

- For each hidden state $h \in C$, there is an “anchor” word $x \in V$ satisfying
  \[ O_{x,g} = 0 \text{ for all } g \neq h. \]

- For comparison, Brown model word class assumption requires every word to be an “anchor”.

Stronger assumption can motivate different algorithmic choices (e.g., clustering normalized rows of left singular vector matrix).
Effect of representation

Data from English treebank, 12 POS tag types

<table>
<thead>
<tr>
<th>Method</th>
<th>dev acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anchor</td>
<td>53.4</td>
</tr>
<tr>
<td>Anchor+CCA</td>
<td>57.0</td>
</tr>
<tr>
<td>Anchor+Rand</td>
<td>48.2</td>
</tr>
<tr>
<td>Spectral+√</td>
<td>66.1</td>
</tr>
</tbody>
</table>

- Anchor = Arora-Ge-Moitra “conditional probability” representation.
- Anchor-CCA = same as Arora-Ge-Moitra, except apply CCA projection to right-hand side (Cohen-Collins, 2014).
- Anchor-Rand = same as Arora-Ge-Moitra, except apply random projection to right-hand side (Ding et al, 2013).